

Instanzbezogene Auswertungen

Anzahl an gestarteten und beendeten Instanzen je Definition

```
SELECT inst1.definitionname AS Prozess, COUNT(DISTINCT(inst1.id)) AS "Anzahl
gestarteter Instanzen", COUNT(DISTINCT(inst2.id)) AS "Anzahl beendeter
Instanzen", COUNT(DISTINCT(task.id)) AS "Anzahl offener Aufgaben"
FROM view_instance inst1 LEFT JOIN view_instance inst2 ON inst1.id=inst2.id
AND inst2.end IS NOT NULL
INNER JOIN view_activity act ON act.instanceid=inst1.id
LEFT JOIN view_task task ON task.instanceid=inst1.id
AND task.end IS NULL AND act.start IS NOT NULL AND act.end IS NULL AND
inst1.end IS NULL AND inst1.archiv IS FALSE
GROUP BY inst1.definitionname
```

Anzahl an gestarteten und beendeten Instanzen sowie die Anzahl an offenen Aufgaben einer Prozessdefinition

```
SELECT inst1.definitionname AS Prozess, COUNT(inst1.id) AS "Anzahl
gestarteter Instanzen", COUNT(inst2.id) AS "Anzahl beendeter Instanzen",
(SELECT COUNT(DISTINCT(task.id))
FROM view_task task INNER JOIN view_instance inst ON task.instanceid =
inst.id INNER JOIN view_activity act ON act.instanceid = inst.id
WHERE task.isopen IS TRUE AND act.END IS NULL AND inst.END IS NULL AND
inst.archiv IS FALSE
AND inst.definitionname = Prozess) AS "Anzahl offene Aufgaben"
FROM view_instance inst1 LEFT JOIN view_instance inst2 ON inst1.id=inst2.id
AND inst2.end IS NOT NULL
WHERE inst1.definitionname = "Name der Prozessdefinition"
```

Anzahl an gestarteten, beendeten, abgebrochenen und laufenden Instanzen je Prozessdefinition

```
SELECT definitionname AS Definitionsname,
SUM(alle) AS "Gestartete Instanzen",
SUM(beendet) AS "Beendete Instanzen (mit Ende)",
SUM(abgebrochen) AS "Abgebrochene Instanzen (Archiviert ohne Ende)",
SUM(laufend) AS "Laufende Instanzen"
FROM (SELECT IF(archiv IS TRUE,
IF(END IS NULL,1,0),0) AS abgebrochen,
IF(END IS NOT NULL,1,0) AS beendet,
IF(END IS NULL,IF(archiv IS TRUE,0,1),0) AS laufend,
1 AS alle, definitionname
FROM view_instance) AS subqu GROUP BY definitionname
```

Finanzielle Aufwände je Instanz

```
SELECT inst.name AS Instanzname, SUM(eff.value) AS Finanzaufwände
FROM view_effort eff
INNER JOIN view_instance inst ON eff.instanceID=inst.id
WHERE eff.effortType="FINANCE"
GROUP BY inst.id
```

Anzahl der Schleifen (>0) je Instanz

```
SELECT inst.name, act.loopcount FROM view_activity act
INNER JOIN view_instance inst ON inst.id=act.instanceID
WHERE act.loopcount IS NOT NULL GROUP BY inst.id
```

Aktuelle Aktivität(en) aller offenen Instanzen

```
SELECT inst.name AS Prozessinstanz, group_concat(act.name SEPARATOR ', ') AS
Aktivität
FROM view_instance inst
INNER JOIN view_activity act ON act.instanceid = inst.id AND act.START IS
NOT NULL AND act.END IS NULL AND act.TYPE = 'K'
WHERE inst.archiv = FALSE AND inst.END IS NULL GROUP BY Prozessinstanz
```

Alle offenen Instanzen mit einer bestimmten Zeichenfolge im Instanznamen

```
SELECT DISTINCT pi.name AS Instanzname, pi.definitionName AS Prozessname
FROM view_activity a, view_instance pi
WHERE a.end IS NULL AND a.start IS NOT NULL AND pi.id = a.instanceId AND
pi.archiv = 0 AND pi.end IS NULL AND pi.name LIKE "%steffen%"
```

Durchschnittliche Durchlaufzeit (in hh:mm:ss) von allen Instanzen aller Prozessdefinitionen je Monat

```
SELECT Monat,
IF((FLOOR(sekunden))<60,
concat(IF((FLOOR(sekunden))<10,"00:00:0","00:00:"),(FLOOR(sekunden))),
IF((FLOOR(sekunden))>3600,
CAST(concat(h,IF(FLOOR((FLOOR(sekunden)-h*3600-
sek)/60)<10,":0",":"),FLOOR((FLOOR(sekunden)-h*3600-
sek)/60),IF(sek<10,":0",":"),sek) AS CHAR),
concat(IF(FLOOR(sekunden/60)<10,"00:0","00:"),FLOOR(sekunden/60),IF(FLOOR(se
k)<10,":0",":"),FLOOR(sek))
)
) AS "Durchschnittliche Durchlaufzeit" FROM
```

```
(SELECT FLOOR(avg(TIMESTAMPDIFF(SECOND,creationtime,END))) AS sekunden,
FLOOR(avg(TIMESTAMPDIFF(SECOND,creationtime,END))%60) AS sek,
FLOOR(avg(TIMESTAMPDIFF(SECOND,creationtime,END))/3600) AS h,
concat(SUBSTRING(monthname(creationtime),1,3),"
",SUBSTRING(CAST(YEAR(creationtime) AS CHAR),3,2)) AS Monat
FROM view_instance WHERE END IS NOT NULL GROUP BY Monat ORDER BY
YEAR(creationtime), MONTH(creationtime) ) AS dusub
```

Durchschnittliche Durchlaufzeit (in h) von allen Instanzen aller Prozessdefinitionen je Monat

```
SELECT Monat, h
AS "Durchschnittliche Durchlaufzeit in Stunden" FROM
(SELECT CAST((avg(TIMESTAMPDIFF(SECOND,creationtime,END))/3600) AS
DECIMAL(10,1)) AS h,
concat(SUBSTRING(monthname(creationtime),1,3),"
",SUBSTRING(CAST(YEAR(creationtime) AS CHAR),3,2)) AS Monat
FROM view_instance WHERE END IS NOT NULL GROUP BY Monat ORDER BY
YEAR(creationtime), MONTH(creationtime) ) AS dusub
```

Durchschnittliche Durchlaufzeit (in h) von allen Instanzen einer Prozessdefinitionen je Monat

```
SELECT Monat, h
AS "Durchschnittliche Durchlaufzeit in Stunden" FROM
(SELECT CAST((avg(TIMESTAMPDIFF(SECOND,creationtime,END))/3600) AS
DECIMAL(10,1)) AS h,
concat(SUBSTRING(monthname(creationtime),1,3),"
",SUBSTRING(CAST(YEAR(creationtime) AS CHAR),3,2)) AS Monat
FROM view_instance WHERE END IS NOT NULL AND definitionname = "Name der
Definition" GROUP BY Monat ORDER BY YEAR(creationtime), MONTH(creationtime) )
AS dusub
```

Instanzen je Aktivität einer Prozessdefinition (analog zur Auswertung in den PM-Clients)

```
SELECT va.name AS "Aktivitäten", COUNT(va.name) AS Anzahl
FROM view_activity va JOIN view_instance vi ON vi.id = va.instanceId
WHERE va.END IS NULL
AND va.START IS NOT NULL
AND va.TYPE = "K"
AND vi.definitionName = "Name der Prozessdefinition"
AND vi.archiv = FALSE
GROUP BY va.name
```

Anzahl der gestarteten und beendeten Instanzen je Monat und Definition

```
SELECT Monat, Prozess, gestartete AS "Anzahl gestartete Instanzen", beendete
AS "Anzahl beendete Instanzen"
FROM
(SELECT COUNT(DISTINCT(inst1.id)) AS gestartete,
concat(SUBSTRING(monthname(inst1.creationtime),1,3), " ",
,SUBSTRING(CAST(YEAR(inst1.creationtime) AS CHAR),3,2)) AS Monat,
inst1.definitionname AS Prozess, COUNT(DISTINCT(inst2.id)) AS beendete,
YEAR(inst1.creationtime) AS jahr, MONTH(inst1.creationtime) AS monat2
FROM view_instance inst1 LEFT OUTER JOIN view_instance inst2 ON
concat(SUBSTRING(monthname(inst1.creationtime),1,3), " ",
,SUBSTRING(CAST(YEAR(inst1.creationtime) AS
CHAR),3,2))=concat(SUBSTRING(monthname(inst2.end),1,3), " ",
,SUBSTRING(CAST(YEAR(inst2.end) AS CHAR),3,2)) AND
inst1.definitionname=inst2.definitionname
GROUP BY Monat, Prozess
UNION
SELECT COUNT(DISTINCT(inst1.id)) AS gestartete,
concat(SUBSTRING(monthname(inst2.end),1,3), " ",
,SUBSTRING(CAST(YEAR(inst2.end) AS CHAR),3,2)) AS Monat,
inst2.definitionname AS Prozess, COUNT(DISTINCT(inst2.id)) AS beendete,
YEAR(inst2.end) AS jahr, MONTH(inst2.end) AS monat2
FROM view_instance inst1 RIGHT OUTER JOIN view_instance inst2 ON
concat(SUBSTRING(monthname(inst1.creationtime),1,3), " ",
,SUBSTRING(CAST(YEAR(inst1.creationtime) AS
CHAR),3,2))=concat(SUBSTRING(monthname(inst2.end),1,3), " ",
,SUBSTRING(CAST(YEAR(inst2.end) AS CHAR),3,2)) AND
inst1.definitionname=inst2.definitionname
WHERE inst2.end IS NOT NULL GROUP BY Monat, Prozess) AS spalten
ORDER BY jahr, monat2, Prozess
```

Minimale, maximale und durchschnittliche Laufzeit (in hh:mm:ss) der Instanzen je Prozessdefinition

```
SELECT name AS Prozess,
IF((FLOOR(avgSekunden))<60,
concat(IF((FLOOR(avgSekunden))<10,"00:00:0","00:00:"),(FLOOR(avgSekunden))),
IF((FLOOR(avgSekunden))>3600,
CAST(concat(avgH,IF(FLOOR((FLOOR(avgSekunden)- avgH*3600-
avgSek)/60)<10,":0",":"),FLOOR((FLOOR(avgSekunden)- avgH*3600-
avgSek)/60),IF(avgSek<10,":0",":"),avgSek) AS CHAR),
concat(IF(FLOOR(avgSekunden/60)<10,"00:0","00:"),FLOOR(avgSekunden/60),IF(FLOOR(avgSek)<10,":0",":"),FLOOR(avgSek))
)
)AS "Durchschnittliche Durchlaufszeit",
IF((FLOOR(minSekunden))<60,
concat(IF((FLOOR(minSekunden))<10,"00:00:0","00:00:"),(FLOOR(minSekunden))),
```

```

    IF((FLOOR(minSekunden))>3600,
        CAST(concat(minH,IF(FLOOR((FLOOR(minSekunden)- minH*3600-
minSek)/60)<10,":0",":"),FLOOR((FLOOR(minSekunden)- minH*3600-
minSek)/60),IF(minSek<10,":0",":"),minSek) AS CHAR),
concat(IF(FLOOR(minSekunden/60)<10,"00:0","00:"),FLOOR(minSekunden/60),IF(FLOOR(minSek)<10,":0",":"),FLOOR(minSek))
    )
) AS "Minimale Durchlaufszeit",
IF((FLOOR(maxSekunden))<60,
concat(IF((FLOOR(maxSekunden))<10,"00:00:0","00:00:"),(FLOOR(maxSekunden))),
    IF((FLOOR(maxSekunden))>3600,
        CAST(concat(maxH,IF(FLOOR((FLOOR(maxSekunden)- maxH*3600-
maxSek)/60)<10,":0",":"),FLOOR((FLOOR(maxSekunden)- maxH*3600-
maxSek)/60),IF(maxSek<10,":0",":"),maxSek) AS CHAR),
concat(IF(FLOOR(maxSekunden/60)<10,"00:0","00:"),FLOOR(maxSekunden/60),IF(FLOOR(maxSek)<10,":0",":"),FLOOR(maxSek))
    )
) AS "Maximale Durchlaufszeit"

FROM
(SELECT definitionname AS name,
FLOOR(MIN((unix_timestamp(inst.END) - unix_timestamp(inst.creationtime))))
AS minSekunden,
FLOOR(MIN((unix_timestamp(inst.END) -
unix_timestamp(inst.creationtime)))%60) AS minSek,
FLOOR(MIN((unix_timestamp(inst.END) -
unix_timestamp(inst.creationtime))/3600)) AS minH,
FLOOR(MAX((unix_timestamp(inst.END) - unix_timestamp(inst.creationtime))))
AS maxSekunden,
FLOOR(MAX((unix_timestamp(inst.END) -
unix_timestamp(inst.creationtime)))%60) AS maxSek,
FLOOR(MAX((unix_timestamp(inst.END) -
unix_timestamp(inst.creationtime))/3600)) AS maxH,
FLOOR(avg((unix_timestamp(inst.END) - unix_timestamp(inst.creationtime))))
AS avgSekunden,
FLOOR(avg((unix_timestamp(inst.END) -
unix_timestamp(inst.creationtime)))%60) AS avgSek,
FLOOR(avg((unix_timestamp(inst.END) -
unix_timestamp(inst.creationtime))/3600)) AS avgH,
(unix_timestamp(inst.END)-unix_timestamp(inst.creationtime)) AS datediff
FROM view_instance inst
WHERE inst.END IS NOT NULL GROUP BY definitionname
) AS datediffsub
WHERE datediff > 0
GROUP BY name

```

Durchschnittliche Durchlaufzeit (in hh:mm:ss) der Instanzen einer bestimmten Prozessdefinition je Monat

```
SELECT Monat,
```

```
IF((FLOOR(sekunden))<60,
  concat(IF((FLOOR(sekunden))<10,"00:00:0","00:00:"),(FLOOR(sekunden))),
  IF((FLOOR(sekunden))>3600,
    CAST(concat(h,IF(FLOOR((FLOOR(sekunden)- h*3600-
sek)/60)<10,":0",":"),FLOOR((FLOOR(sekunden)- h*3600-
sek)/60),IF(sek<10,":0",":"),sek) AS CHAR),
concat(IF(FLOOR(sekunden/60)<10,"00:0","00:"),FLOOR(sekunden/60),IF(FLOOR(se
k)<10,":0",":"),FLOOR(sek))
)
) AS "Durchschnittliche Durchlaufzeit" FROM
(SELECT FLOOR(avg(TIMESTAMPDIFF(SECOND,vi.creationtime,vi.END))) AS
sekunden,
FLOOR(avg(TIMESTAMPDIFF(SECOND,vi.creationtime,vi.END))%60) AS sek,
FLOOR(avg(TIMESTAMPDIFF(SECOND,vi.creationtime,vi.END))/3600) AS h,
concat(SUBSTRING(monthname(vi.creationTime),1,3),"
",SUBSTRING(CAST(YEAR(vi.creationTime) AS CHAR),3,2)) AS Monat,
vi.creationtime AS creationtime
FROM view_instance vi, view_definition vd
WHERE vd.id = vi.definitionId AND vi.END IS NOT NULL AND
vd.name = "Name der Prozessdefinition" GROUP BY YEAR(vi.creationtime),
MONTH(vi.creationtime)) AS dusub
GROUP BY YEAR(creationtime), MONTH(creationtime)
```

Aufgabenbezogenen Auswertungen

Alle offenen Aufgaben eines bestimmten Users

```
SELECT i.lastname AS Nachname, i.firstname AS Vorname, i.name AS Username,
t.name AS Task, pi.name AS Instanzname, pi.definitionName AS Prozessname
FROM view_activity a, view_task t, view_identity i, view_instance pi
WHERE a.id = t.activity AND t.actor = i.id AND a.end IS NULL AND t.end IS
NULL AND pi.id = a.instanceId AND pi.archiv = 0 AND pi.end IS NULL AND
i.lastname = "Barth"
```

Anzahl an erstellten, erledigten und offenen Aufgaben je Prozessdefinition

```
SELECT inst.definitionname AS Prozess,
COUNT(t1.id) AS "Anzahl erstellte Aufgaben",
COUNT(t2.id) AS "Anzahl erledigte Aufgaben",
COUNT(t3.id) AS "Anzahl offene Aufgaben"
FROM view_task t1 INNER JOIN view_activity act ON t1.activity=act.id
INNER JOIN view_instance inst ON t1.instanceid = inst.id
LEFT JOIN view_task t2 ON t1.id=t2.id AND (t2.end IS NOT NULL OR act.END IS
NOT NULL)
LEFT JOIN view_task t3 ON t1.id=t3.id AND t3.end IS NULL AND inst.end IS
NULL AND inst.archiv IS FALSE
```

```
GROUP BY inst.definitionname
```

Anzahl an erstellten, erledigten und offenen Aufgaben einer bestimmten Prozessdefinition

```
SELECT inst.definitionname AS Prozess,
COUNT(t1.id) AS "Anzahl erstellte Aufgaben",
COUNT(t2.id) AS "Anzahl erledigte Aufgaben",
COUNT(t3.id) AS "Anzahl offene Aufgaben"
FROM view_task t1 INNER JOIN view_activity act ON t1.activity=act.id INNER
JOIN view_instance inst ON t1.instanceid = inst.id
LEFT JOIN view_task t2 ON t1.id=t2.id AND (t2.end IS NOT NULL OR act.END IS
NOT NULL)
LEFT JOIN view_task t3 ON t1.id=t3.id AND t3.end IS NULL AND inst.end IS
NULL AND inst.archiv IS FALSE
WHERE inst.definitionname ="Name der Prozessdefinition"
```

Durchschnittliche Zeit (in hh:mm:ss) von Erstellung bis Erledigung von allen Aufgaben (egal von welcher Prozessdefinition und auch außerhalb der Arbeitszeit)

```
SELECT IF((FLOOR(sekunden))<60,
concat(IF((FLOOR(sekunden))<10,"00:00:0","00:00:"),(FLOOR(sekunden))),
IF((FLOOR(sekunden))>3600,
CAST(concat(h,IF(FLOOR((FLOOR(sekunden)- h*3600-
sek)/60)<10,":0",":"),FLOOR((FLOOR(sekunden)- h*3600-
sek)/60),IF(sek<10,":0",":"),sek) AS CHAR),
concat(IF(FLOOR(sekunden/60)<10,"00:0","00:"),FLOOR(sekunden/60),IF(FLOOR(se
k)<10,":0",":"),FLOOR(sek))
)
) AS "Durchschnittliche Liegezeit von Aufgaben" FROM (SELECT
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))) AS sekunden,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))%60) AS sek,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))/3600) AS h
FROM view_task WHERE END IS NOT NULL) AS dusub
```

Durchschnittliche Zeit (in hh:mm:ss) von Erstellung bis Erledigung von allen Aufgaben je Monat (egal von welcher Prozessdefinition und auch außerhalb der Arbeitszeit)

```
SELECT Monat, IF((FLOOR(sekunden))<60,
concat(IF((FLOOR(sekunden))<10,"00:00:0","00:00:"),(FLOOR(sekunden))),
IF((FLOOR(sekunden))>3600,
CAST(concat(h,IF(FLOOR((FLOOR(sekunden)- h*3600-
sek)/60)<10,":0",":"),FLOOR((FLOOR(sekunden)- h*3600-
sek)/60),IF(sek<10,":0",":"),sek) AS CHAR),
concat(IF(FLOOR(sekunden/60)<10,"00:0","00:"),FLOOR(sekunden/60),IF(FLOOR(se
k)<10,":0",":"),FLOOR(sek))
)
) AS "Durchschnittliche Liegezeit von Aufgaben" FROM
```



```
(SELECT concat(SUBSTRING(monthname(START),1,3),"  
",SUBSTRING(CAST(YEAR(START) AS CHAR),3,2)) AS Monat,  
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))) AS sekunden,  
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))%60) AS sek,  
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))/3600) AS h FROM  
view_task  
WHERE isOpen = "0" GROUP BY YEAR(START), MONTH(START)) AS dusub
```

Durchschnittliche Zeit (in hh:mm:ss) von Erstellung bis Erledigung von allen Aufgaben je Monat ohne Ausreißer (egal von welcher Prozessdefinition und auch außerhalb der Arbeitszeit)

```
SELECT Monat, IF((FLOOR(sekunden))<60,  
concat(IF((FLOOR(sekunden))<10,"00:00:0","00:00:"),(FLOOR(sekunden))),  
IF((FLOOR(sekunden))>3600,  
CAST(concat(h,IF(FLOOR((FLOOR(sekunden)- h*3600-  
sek)/60)<10,":0",":"),FLOOR((FLOOR(sekunden)- h*3600-  
sek)/60),IF(sek<10,":0",":"),sek) AS CHAR),  
concat(IF(FLOOR(sekunden/60)<10,"00:0","00:"),FLOOR(sekunden/60),IF(FLOOR(sek  
k)<10,":0",":"),FLOOR(sek))  
)  
) AS "Durchschnittliche Liegezeit von Aufgaben" FROM  
(SELECT concat(SUBSTRING(monthname(START),1,3),"  
",SUBSTRING(CAST(YEAR(START) AS CHAR),3,2)) AS Monat,  
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))) AS sekunden,  
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))%60) AS sek,  
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))/3600) AS h FROM  
view_task  
WHERE isOpen = "0" AND CAST(FLOOR((unix_timestamp(END) -  
unix_timestamp(START))) AS DECIMAL(10,10)) < 10000 GROUP BY YEAR(START),  
MONTH(START)) AS dusub
```

Alle offenen Aufgaben mit aktiver Instanz (nicht beendet und nicht archiviert) und aktiver Aktivität (nicht beendet und nicht archiviert) mit Bearbeiter

```
SELECT tsk.name AS Aufgabe, IF (concat(ident.firstname, ",",ident.lastname)  
IS NOT NULL,  
concat(ident.firstname, ",",ident.lastname), grouped.name) AS 'Bearbeiter',  
inst.name AS 'Instanz', def.name AS 'Prozess', tsk.creationTime AS START  
FROM view_task tsk  
LEFT JOIN view_identity ident ON tsk.actor = ident.id  
LEFT JOIN view_identity grouped ON tsk.pooledActor = grouped.id  
INNER JOIN view_instance inst ON tsk.instanceId = inst.id  
INNER JOIN view_definition def ON inst.definitionId = def.id  
INNER JOIN view_activity act ON tsk.activity = act.id  
WHERE inst.archiv = 'false' AND tsk.isOpen = 1
```


Durchschnittliche, minimale und maximale Durchlaufzeit aller Aufgaben einer bestimmten Prozessdefinition (in hh:mm:ss)

```

SELECT name AS Aufgabe,
IF((FLOOR(avgSekunden))<60,
concat(IF((FLOOR(avgSekunden))<10,"00:00:0","00:00:"),(FLOOR(avgSekunden))),
IF((FLOOR(avgSekunden))>3600,
CAST(concat(avgH,IF(FLOOR((FLOOR(avgSekunden)- avgH*3600-
avgSek)/60)<10,":0",":"),FLOOR((FLOOR(avgSekunden)- avgH*3600-
avgSek)/60),IF(avgSek<10,":0",":"),avgSek) AS CHAR),
concat(IF(FLOOR(avgSekunden/60)<10,"00:0","00:"),FLOOR(avgSekunden/60),IF(FLOOR(avgSek)<10,":0",":"),FLOOR(avgSek))
)
)AS "Durchschnittliche Durchlaufzeit",
IF((FLOOR(minSekunden))<60,
concat(IF((FLOOR(minSekunden))<10,"00:00:0","00:00:"),(FLOOR(minSekunden))),
IF((FLOOR(minSekunden))>3600,
CAST(concat(minH,IF(FLOOR((FLOOR(minSekunden)- minH*3600-
minSek)/60)<10,":0",":"),FLOOR((FLOOR(minSekunden)- minH*3600-
minSek)/60),IF(minSek<10,":0",":"),minSek) AS CHAR),
concat(IF(FLOOR(minSekunden/60)<10,"00:0","00:"),FLOOR(minSekunden/60),IF(FLOOR(minSek)<10,":0",":"),FLOOR(minSek))
)
) AS "Minimale Durchlaufzeit",
IF((FLOOR(maxSekunden))<60,
concat(IF((FLOOR(maxSekunden))<10,"00:00:0","00:00:"),(FLOOR(maxSekunden))),
IF((FLOOR(maxSekunden))>3600,
CAST(concat(maxH,IF(FLOOR((FLOOR(maxSekunden)- maxH*3600-
maxSek)/60)<10,":0",":"),FLOOR((FLOOR(maxSekunden)- maxH*3600-
maxSek)/60),IF(maxSek<10,":0",":"),maxSek) AS CHAR),
concat(IF(FLOOR(maxSekunden/60)<10,"00:0","00:"),FLOOR(maxSekunden/60),IF(FLOOR(maxSek)<10,":0",":"),FLOOR(maxSek))
)
) AS "Maximale Durchlaufzeit"
FROM
(SELECT inst.definitionname AS defName, task.name AS name,
FLOOR(MIN((unix_timestamp(task.END) - unix_timestamp(task.START)))) AS
minSekunden,
FLOOR(MIN((unix_timestamp(task.END) - unix_timestamp(task.START)))%60) AS
minSek,
FLOOR(MIN((unix_timestamp(task.END) - unix_timestamp(task.START))/3600)) AS
minH,
FLOOR(MAX((unix_timestamp(task.END) - unix_timestamp(task.START)))) AS
maxSekunden,
FLOOR(MAX((unix_timestamp(task.END) - unix_timestamp(task.START)))%60) AS
maxSek,
FLOOR(MAX((unix_timestamp(task.END) - unix_timestamp(task.START))/3600)) AS
maxH,
FLOOR(avg((unix_timestamp(task.END) - unix_timestamp(task.START)))) AS
avgSekunden,

```

```
FLOOR(avg((unix_timestamp(task.END) - unix_timestamp(task.START))%60) AS
avgSek,
FLOOR(avg((unix_timestamp(task.END) - unix_timestamp(task.START))/3600)) AS
avgH,
(unix_timestamp(task.END)-unix_timestamp(task.START)) AS datediff
FROM view_task task INNER JOIN view_instance inst ON task.instanceId=inst.id
WHERE task.END IS NOT NULL AND task.START IS NOT NULL
GROUP BY task.name
) AS datediffsub
WHERE datediff > 0
AND defName = "Name der Prozessdefinition"
GROUP BY name
```

Aktivitätenbezogene Auswertungen

Anzahl an erstellten, beendeten und offenen Aktivitäten je Prozessdefinition

```
SELECT inst.definitionname AS Prozess, COUNT(DISTINCT(act1.id)) AS "Anzahl
erstellte Aktivitäten", COUNT(DISTINCT(act2.id)) AS "Anzahl beendete
Aktivitäten", COUNT(task.id) AS "Anzahl offene Aufgaben"
FROM view_activity act1 INNER JOIN view_instance inst ON
act1.instanceid=inst.id AND act1.START IS NOT NULL AND TYPE IN ("K","S")
LEFT JOIN view_task task ON act1.id=task.activity AND task.END IS NULL AND
task.START IS NOT NULL
LEFT JOIN view_activity act2 ON act1.id=act2.id AND act2.END IS NOT NULL
GROUP BY Prozess
```

Anzahl an erstellten, beendeten und offenen Aktivitäten einer bestimmten Prozessdefinition

```
SELECT defName AS Prozess, COUNT(ended) AS "Anzahl erstellte Aktivitäten",
SUM(ended) AS "Anzahl beendete Aktivitäten", (SELECT
COUNT(DISTINCT(task.id))
FROM view_task task INNER JOIN view_instance inst ON task.instanceid =
inst.id INNER JOIN view_activity act ON act.instanceid = inst.id
WHERE task.isopen IS TRUE AND act.END IS NULL AND inst.END IS NULL AND
inst.archiv IS FALSE
AND inst.definitionname = defName) AS "Anzahl offene Aktivitäten" FROM
(SELECT inst.definitionname AS defname, IF(act.END IS NULL,0,1) AS ended
FROM view_activity act
INNER JOIN view_instance inst ON inst.id=act.instanceid
WHERE TYPE IN ("K","S") AND START IS NOT NULL AND inst.definitionname= "Name
der Prozessdefinition") AS subQu
```

Durchschnittliche Zeit (in hh:mm:ss) von Erstellung bis Erledigung von allen Aktivitäten (egal von welcher Prozessdefinition und auch außerhalb der Arbeitszeit)

```
SELECT IF((FLOOR(sekunden))<60,
  concat(IF((FLOOR(sekunden))<10,"00:00:0","00:00:"),(FLOOR(sekunden))),
  IF((FLOOR(sekunden))>3600,
    CAST(concat(h,IF(FLOOR((FLOOR(sekunden)- h*3600-
sek)/60)<10,":0",":"),FLOOR((FLOOR(sekunden)- h*3600-
sek)/60),IF(sek<10,":0",":"),sek) AS CHAR),
concat(IF(FLOOR(sekunden/60)<10,"00:0","00:"),FLOOR(sekunden/60),IF(FLOOR(se
k)<10,":0",":"),FLOOR(sek))
)
) AS "Durchschnittliche Liegezeit von Aktivitäten" FROM (SELECT
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))) AS sekunden,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START))%60) AS sek,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START))/3600) AS h FROM
view_activity
WHERE START IS NOT NULL AND END IS NOT NULL AND TYPE IN ("K","S")) AS dusub
```

Durchschnittliche Zeit (in hh:mm:ss) von Erstellung bis Erledigung von allen Aktivitäten je Monat (egal von welcher Prozessdefinition und auch außerhalb der Arbeitszeit) je Monat

```
SELECT Monat, IF((FLOOR(sekunden))<60,
  concat(IF((FLOOR(sekunden))<10,"00:00:0","00:00:"),(FLOOR(sekunden))),
  IF((FLOOR(sekunden))>3600,
    CAST(concat(h,IF(FLOOR((FLOOR(sekunden)- h*3600-
sek)/60)<10,":0",":"),FLOOR((FLOOR(sekunden)- h*3600-
sek)/60),IF(sek<10,":0",":"),sek) AS CHAR),
concat(IF(FLOOR(sekunden/60)<10,"00:0","00:"),FLOOR(sekunden/60),IF(FLOOR(se
k)<10,":0",":"),FLOOR(sek))
)
) AS "Durchschnittliche Liegezeit von Aktivitäten" FROM
(SELECT concat(SUBSTRING(monthname(START),1,3),"
",SUBSTRING(CAST(YEAR(START) AS CHAR),3,2)) AS Monat,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))) AS sekunden,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START))%60) AS sek,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START))/3600) AS h FROM
view_activity WHERE END IS NOT NULL
AND START IS NOT NULL AND TYPE IN ("K","S") GROUP BY Monat ORDER BY
YEAR(START), monthname(START)) AS dusub
```

Durchschnittliche Zeit (in hh:mm:ss) von Erstellung bis Erledigung von allen Aktivitäten je Monat ohne Ausreißer (von allen Prozessdefinitionen und auch außerhalb der Arbeitszeit)

```
SELECT Monat, IF((FLOOR(sekunden))<60,
  concat(IF((FLOOR(sekunden))<10,"00:00:0","00:00:"),(FLOOR(sekunden))),
  IF((FLOOR(sekunden))>3600,
```

```
CAST(concat(h,IF(FLOOR((FLOOR(sekunden)- h*3600-
sek)/60)<10,":0",":"),FLOOR((FLOOR(sekunden)- h*3600-
sek)/60),IF(sek<10,":0",":"),sek) AS CHAR),
concat(IF(FLOOR(sekunden/60)<10,"00:0","00:"),FLOOR(sekunden/60),IF(FLOOR(se
k)<10,":0",":"),FLOOR(sek))
)
) AS "Durchschnittliche Liegezeit von Aktivitäten" FROM
(SELECT concat(SUBSTRING(monthname(START),1,3),"
",SUBSTRING(CAST(YEAR(START) AS CHAR),3,2)) AS Monat,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))) AS sekunden,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START))%60) AS sek,
FLOOR(avg((unix_timestamp(END) - unix_timestamp(START)))/3600) AS h FROM
view_activity
WHERE START IS NOT NULL AND END IS NOT NULL AND TYPE IN ("K","S") AND
CAST(FLOOR((unix_timestamp(END) - unix_timestamp(START))) AS DECIMAL(10,10))
< 10000 GROUP BY YEAR(START), monthname(START)) AS dusub
```

Durchschnittliche, minimale und maximale Durchlaufzeit aller Aktivitäten einer bestimmten Prozessdefinition (in hh:mm:ss)

```
SELECT name AS Aktivität,
IF((FLOOR(avgSekunden))<60,
concat(IF((FLOOR(avgSekunden))<10,"00:00:0","00:00:"),(FLOOR(avgSekunden))),
IF((FLOOR(avgSekunden))>3600,
CAST(concat(avgH,IF(FLOOR((FLOOR(avgSekunden)- avgH*3600-
avgSek)/60)<10,":0",":"),FLOOR((FLOOR(avgSekunden)- avgH*3600-
avgSek)/60),IF(avgSek<10,":0",":"),avgSek) AS CHAR),
concat(IF(FLOOR(avgSekunden/60)<10,"00:0","00:"),FLOOR(avgSekunden/60),IF(FLOOR(avgSek)<10,":0",":"),FLOOR(avgSek))
)
)AS "Durchschnittliche Durchlaufzeit",
IF((FLOOR(minSekunden))<60,
concat(IF((FLOOR(minSekunden))<10,"00:00:0","00:00:"),(FLOOR(minSekunden))),
IF((FLOOR(minSekunden))>3600,
CAST(concat(minH,IF(FLOOR((FLOOR(minSekunden)- minH*3600-
minSek)/60)<10,":0",":"),FLOOR((FLOOR(minSekunden)- minH*3600-
minSek)/60),IF(minSek<10,":0",":"),minSek) AS CHAR),
concat(IF(FLOOR(minSekunden/60)<10,"00:0","00:"),FLOOR(minSekunden/60),IF(FLOOR(minSek)<10,":0",":"),FLOOR(minSek))
)
) AS "Minimale Durchlaufzeit",
IF((FLOOR(maxSekunden))<60,
concat(IF((FLOOR(maxSekunden))<10,"00:00:0","00:00:"),(FLOOR(maxSekunden))),
IF((FLOOR(maxSekunden))>3600,
CAST(concat(maxH,IF(FLOOR((FLOOR(maxSekunden)- maxH*3600-
maxSek)/60)<10,":0",":"),FLOOR((FLOOR(maxSekunden)- maxH*3600-
maxSek)/60),IF(maxSek<10,":0",":"),maxSek) AS CHAR),
concat(IF(FLOOR(maxSekunden/60)<10,"00:0","00:"),FLOOR(maxSekunden/60),IF(FLOOR(maxSek)<10,":0",":"),FLOOR(maxSek))
)
) AS "Maximale Durchlaufzeit"
```

```

)
) AS "Maximale Durchlaufzeit"

FROM
(SELECT activity.name AS name, definitionname AS defName,
FLOOR(MIN((unix_timestamp(activity.END) - unix_timestamp(activity.START))))
AS minSekunden,
FLOOR(MIN((unix_timestamp(activity.END) -
unix_timestamp(activity.START)))%60) AS minSek,
FLOOR(MIN((unix_timestamp(activity.END) -
unix_timestamp(activity.START))/3600)) AS minH,
FLOOR(MAX((unix_timestamp(activity.END) - unix_timestamp(activity.START))))
AS maxSekunden,
FLOOR(MAX((unix_timestamp(activity.END) -
unix_timestamp(activity.START)))%60) AS maxSek,
FLOOR(MAX((unix_timestamp(activity.END) -
unix_timestamp(activity.START))/3600)) AS maxH,
FLOOR(avg((unix_timestamp(activity.END) - unix_timestamp(activity.START))))
AS avgSekunden,
FLOOR(avg((unix_timestamp(activity.END) -
unix_timestamp(activity.START)))%60) AS avgSek,
FLOOR(avg((unix_timestamp(activity.END) -
unix_timestamp(activity.START))/3600)) AS avgH,
(unix_timestamp(activity.END)-unix_timestamp(activity.START)) AS datediff
FROM view_activity activity INNER JOIN view_instance inst ON
activity.instanceId=inst.id
WHERE activity.END IS NOT NULL AND activity.START IS NOT NULL AND
activity.TYPE IN ("K","S")
GROUP BY activity.name
) AS datediffsub
WHERE datediff > 0
AND defName = "Name der Prozessdefinition"
GROUP BY name

```

Definitionsbezogene Auswertungen

Anzahl der Schleifen (>0) je Definition

```

SELECT inst.definitionname AS Prozess, SUM(act.loopcount) AS
Schleifendurchläufe FROM view_activity act
INNER JOIN view_instance inst ON inst.id=act.instanceID
WHERE act.loopcount IS NOT NULL
GROUP BY inst.definitionname

```

Variablenbezogene Auswertungen

Prozessvariablen-Template

```
INNER JOIN
(
SELECT var1.stringvalue AS `1`, var2.stringvalue AS `2`, var3.stringvalue AS
`3`, inst.ID
FROM view_instance inst
LEFT JOIN view_variable var1 ON inst.ID = var1.instanceID AND var1.name = ""
LEFT JOIN view_variable var2 ON inst.ID = var2.instanceID AND var2.name = ""
LEFT JOIN view_variable var3 ON inst.ID = var3.instanceID AND var3.name = ""
WHERE inst.definitionname = "" AND inst.archiv = "" AND inst.end IS NOT NULL
) AS Variablen1 ON Variablen1.ID = Variablen2.ID
```

From:
<https://wiki.tim-solutions.de/> - **TIM Wiki** / [NEW TIM 6 Documentation](#)

Permanent link:
<https://wiki.tim-solutions.de/doku.php?id=software:dashboard:analyses&rev=1386863072>

Last update: **2021/07/01 10:00**

